A Fast Reconstruction Algorithm for Gene Networks

Lorenzo Farina and Ilaria Mogno
Dipartimento di Informatica e Sistemistica "A. Ruberti"
Università degli Studi di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
e-mail: lorenzo.farina@uniroma1.it, mogno@dis.uniroma1.it
tel: +39-0644585690, +39-0644585938
fax: +39-0644585367

February 9, 2008

Abstract

This paper deals with gene networks whose dynamics is assumed to be generated by a continuous–time, linear, time invariant, finite dimensional system (*LTI*) at steady state. In particular, we deal with the problem of network reconstruction in the typical practical situation in which the number of available data is largely insufficient to uniquely determine the network. In order to try to remove this ambiguity, we will exploit the biologically *a priori* assumption of network sparseness, and propose a new algorithm for network reconstruction having a very low computational complexity (linear in the number of genes) so to be able to deal also with very large networks (say, thousands of genes). Its performances are also tested both on artificial data (generated with linear models) and on real data obtained by *Gardner et al.* from the SOS pathway in *Escherichia coli.*

Keywords: Functional Genomics, Computational Biology, Gene Networks, Reverse Engineering, Modelling.

1 Introduction

In this paper we will consider the most simple – not trivial dynamical model of a gene network, define a *reverse engineering problem*[1] and propose a fast algorithm to tackle this problem. We will deal then with continuous–time, linear, time invariant, finite dimensional system (*LTI systems*) at steady state, in view of the relevant biological literature (see, for example [1]). Even though this oversimplified model may not be very realistic "far" from steady state, nevertheless it is a fundamental tool for studying and gaining insight into the basic mechanism that makes this problem an hard one, thus providing a valuable *in numero* testbed for gene networks reconstruction algorithms. In fact, starting from the simplest and then moving toward the more and more complex is a typical scenario in the applied sciences. Consequently, in this paper we will consider only experiments regarding steady state measurements due to "small" input constant perturbations, so to retain linearity of the model.

More precisely, we deal with the problem of reconstructing a gene network in the typical practical situation in which the number of available data is largely insufficient to uniquely determine the network. In order to try to remove this ambiguity, we will exploit some additional biologically relevant *a priori* assumptions such as sparseness, as it will be expalined in the following.

The proposed algorithm has the major advantage to be very fast. In fact, its complexity is $O(2NM^2)$ and consequently it is particularly useful for large networks (large N) and few data (small M).

Finally we will also generate artificial data for the reconstruction algorithm. By doing so, one exactly knows the "true" gene regulation network and then algorithms performance can be evaluated. Moreover, the algorithm will be successfully applied to real data collected from experiments regarding a nine-gene subnetwork of the SOS pathway in *Escherichia coli* and very recently presented in [2].

[1] In the system and control community, this problem is commonly defined as an *identification* problem. We prefer here to adopt the terminology used in the biological literature.

2 Dynamic model of expression data and problem formulation

The idea of modelling gene expression data with differential equations has been explored by a considerable numbers of authors (see for example references [3], [4], [1], [5], [6], [7], [8] and, for a literature review, the interested reader may refer to [9]). Differential equations are used to model gene interactions under the assumption that the rate of change over time of each gene expression level is a function of the expression level of some (usually a few) other genes. Such modeling assumption is based on the reaction kinetics at the biochemical level. A fully realistic model should consider a number of relevant biological issues such as, for example, internal and external noise, time delays, specific classes of nonlinearities, and should also consider the relationships between mRNA and protein concentrations since only the first one is actually measured by microarrays. Clearly, a lot of work remains to be done in the field of gene interaction modelling. Nevertheless, a very simple linear time invariant model has proved to be useful in a number of cases (as in [4] for rat cervical spinal cord data) even if it is clear that nonlinearity is an unavoidable issue since it reflects also the nature of biochemical interactions. However, as in [2], we will assume the system to behave linearly around the steady states points reached in experiments. According to the work of Yeung *et al.* [1] we consider the LTI system described by the following differential equations:

$$\dot{x}_i\left(t\right) = -\lambda_i x_i\left(t\right) + \sum_{j=1}^{N} W_{ij} x_j\left(t\right) + b_i\left(t\right) + \xi_i\left(t\right) \tag{1}$$

for $i = 1, 2, ..., N$, where the state variables x_i's are the concentration of mRNA measured as a difference from the equilibrium state preceding the stimulus[2], the λ_i's are the self–degradation rates, the b_i's are the external stimuli (depending on the specific experiment performed), and the ξ_i's represent (internal) noise. The elements of the matrix W describe the type and strength of the "influence" of the j-th gene on the i-th gene with a positive, zero or negative sign indicating activation, no interaction and repression respectively. As described in [2], an experiment consists in applying

[2] Clearly, these numbers can be positive, negative or zero provided that the measured concentration levels are greater, less or equal to the concentration present prior to the stimulus.

a prescribed (*i.e.* known) stimulus $b_i\,(t)$ which is a persistent perturbation (ideally a step function $\delta_1(t)$ of "amplitude" b_i, *i.e.* $b_i\,(t) = b_i\delta_1(t)$) to M input separately and then use a microarray to measure the response at steady state. Finally, the above situation, in view of equations (1) and taking into account the measurement noise w introduced by the microarray device, can be formally described in the usual compact form as follows:

$$\begin{aligned} \dot{x} &= Ax + B + \xi \\ y &= x + w \end{aligned} \tag{2}$$

where the matrix A is a $N \times N$ matrix which incorporates both self–degradation rates (on its main diagonal entries) and the strength of the gene–to–gene interaction (on its off diagonal entries) and the columns of the $N \times M$ matrix B are the b_i's. We will assume standard normality properties on zero mean noises.

Steady state (equilibria) solutions are given by $Ax_e + Bu_e + \xi = 0 = Ay_e + Bu_e + \xi - Aw$, and if we repeat the M measures a sufficiently large number of times, we can calculate average values and then solve $A\overline{y} + B\overline{u} = 0$, where $E[y_e] = \overline{y}$, $E[u_e] = \overline{u}$ and $E[\xi - Aw] = 0$ so that $E[w] = E[\xi] = 0$.

We can now state the problem considered in [2]:

An Identification (Reverse Engineering) Problem for Gene Networks

Given a network described by (2), initially at rest composed of N genes, where M independent constant inputs[3] *are applied and M noisy measurements y^j are taken at steady state. This process is repeated a "sufficiently" large number of times so that we can average the data and obtain the data matrix $\bar{Y}_{N\times M}$, whose generic element is $\bar{y}_i^j$ where superscripts denote experiments (which are then repeated M times), the overbar denotes averaging and the subscripts denote individual genes.*

Then, a ***reverse engineering problem*** *consists in finding the "best" matrix A such that*

$$A_{N\times N}\bar{Y}_{N\times M} + B_{N\times M} = 0_{N\times M} \tag{3}$$

[3] The matrix B is known and full rank, possibly diagonal.

It is important to note that the matrix A in equation (3) is unique if and only if $M \geq N$, *i.e.* provided that the number of experiments is equal or greater than the number of genes in the network. In what follows we will assume the typical situation in which $M \ll N$ so that equation (3) is underdetermined, that is, it has many solutions yielding $A\bar{Y}_{N\times M} + B = 0$. Thus, in order to find the *best* choice for the matrix A, some *a priori* information has to be exploited incorporating some biological knowledge into the model at hand. One possibility, as discussed in a previous section, is to try to impose the additional biological constraint that usually gene networks are sparse, *i.e.* that generally each gene interacts with only a small percentage of all the genes in the entire genome (see for example [10]). We will therefore exploit this feature and present our algorithm in the following section.

In the next paragraph, we will first address the problem of generating artificial data consistent with the above biological assumption so to be able, after presenting a new reverse engineering algorithm, to evaluate its performances on such generated artificial data and, afterwards, also on real data taken from [2] regarding the SOS pathway of *Escherichia coli.*

3 The artificial data generation

In this section we will briefly describe how one can set up an artificial problem and the data generated by this procedure. First of all we generated data assuming a linear model (as in (1)) and the algorithms were tested on such data. However, in this preliminary work we just wanted to evaluate the performances of the algorithms presented in the next paragraphs. We would like also to stress that we applied our reconstruction algorithms to artificial data, before using real data. The reason simply being that, by doing so, one exactly knows the "true" regulation network, and the algorithm performances can be evaluated.

Consequently, in our first artificial experiments, we generate the data with the linear model described in equation (1). Each repetition of the i–th experiment consists, in this model, in applying a constant unitary input (normalized) and then in picking a sample of the trajectory generated by equations (1) at steady state. We generate then a *sparse*, *asymptotically stable* matrix A, and, as a second step, also the entries of input matrix B for each experiment. Finally, we obtain the artificial data by considering steady state values, and by repeating this process M times, we get the simulated data matrix $\bar{Y}_{N\times M}$[4].

4 A reconstruction algorithm for sparse gene networks

As discussed in the previous sections, the reverse engineering problem for gene networks when the number of measurement is (much) less than the number of genes involved in the network, leads to many alternative networks among which we have to find a way to choosing the "best" one according to a prespecified cost function.

As stated by the reverse engineering problem, once we have collected the mRNA abundance steady state measurements $\bar{Y}_{N\times M}$ from microarrays with $M < N$, we may first find one of the many solutions A of $A\bar{Y} + B = 0$. A standard procedure for the case of interest with $M < N$, is to perform a *singular value decomposition* on the data matrix $\bar{Y}_{N\times M}$ and obtain the matrix A with the smallest L_2 norm. The procedure goes as follows. We decomposed the data matrix $\bar{Y}_{N\times M}$ as $\bar{Y}_{N\times M} = U_{N\times N}S_{N\times M}W^T_{M\times M}$, where $U_{N\times N}$ and $W^T_{M\times M}$ are unitary matrices and the entries σ_{ij} of the matrix $S_{N\times M}$ are such that $\sigma_{ij} = 0$ for all $i \neq j$ and $\sigma_{11} \geq \sigma_{22} \geq \ldots \geq \sigma_{MM} > 0$. The numbers $\sigma_{ii} := \sigma_i$ are the nonnegative square roots of the eigenvalues of $Y_{N\times M}Y^T_{N\times M}$ known as the *singular values* of $Y_{N\times M}$. As previously stated, the SVD provide a simple way to find a solution A to problem (3) with minimal L_2 norm, *i.e.* with minimal $\|A\|_2$, as[5] $A_{svd} = -BW diag_{i=1,\ldots,M}(\frac{1}{\sigma_i})U^T$. All the solutions to (3) are given by $A = A_{svd} + C$,where C is any $N \times N$ matrix satisfying $C_{N\times M}\bar{Y}_{N\times M} = 0$, as one can easily verify by direct substitution. Equivalently, the matrix C is such that the columns of C^T belong to the null space of $\bar{Y}^T_{N\times M}$.

A possible way of "reasonably" choosing the "best" matrix A without evaluating all the feasible solutions, is given next. Our aim is to take into account the sparseness assumption, that is try to incorporate information on the structure (zero pattern) of the matrix A. In particular, we will concentrate on the case in which the number k of nonzero entries in each row of A is not greater than the number of available measurements, *i.e.* $k \leq M$. Con-

[4] As previously stated, the expression level of the genes are calculated as the difference between the current expression level (the cellular concentration of the gene product).

[5] Note that, in this case, we have $A_{svd}\bar{Y}_{N\times M} + B = 0$.

sequently, we have that $k \leq M < N$ with $k \ll N$. The proposed procedure, hereafter formally described, consists of three steps:

The Fast Reverse Engineering Algorithm

1. Assuming $k \leq M < N$ and using the full rank data matrix $\bar{Y}_{N\times M}$, find the minimal L_2 norm solution $A_{svd} = -BW diag_{i=1,\ldots,M}(\frac{1}{\sigma_i})U^T$ to equation $A\bar{Y}_{N\times M} + B = 0$ using SVD of the data matrix $Y_{N\times M}$.
2. For each of the N rows of A_{svd}, determine the smallest in magnitude $N - M + \eta$ with $\eta \geq 1$ entries and consider them as "zero".
3. Taking into account the zero pattern detected at the previous step, find row by row the unique (sparse) matrix A solution of min arg $\left\|A\bar{Y}_{N\times M} + B\right\|$, using the pseudo-inverse operator on the selected submatrix, and we are done.

Here, the basic idea is to exploit the fact that, at the first step, the computed matrix A_{svd} is that of minimal L_2 norm, so that it is *reasonable* to assume that the smallest magnitude entries correspond to zeros in the "true" matrix A. The main advantage of this approach is its simplicity and low computational complexity as it will discussed later in the paper. It's worth noting that Step 3 requires only that, for each of the N rows of A, an $(M - \eta) \times M$ matrix to be pseudo-inverted and this can be done in a very efficient computational way. The complexity of the algorithm is then $O(2NM^2)$ so it is particularly useful for large networks (large N) and few data (small M).

We compare here this algorithm with another one which makes use of an heuristics to find the zero pattern for each row of the matrix. Instead of applying Step 3 (of the *Fast* algorithm), for each row i, we put to zero one entry, say a_{ij}, of the matrix A_{svd} and denote this modified matrix by $A_{svd,ij}$. We evaluate then the error index $\delta_{ij} = \|A_{svd,ij}Y + B\|_2$. The position of the first zero in the i-th row is then that corresponding to the minimal value of δ_{ij}. We repeat this process until we have placed all the $N - M + \eta$ zeros. More precisely, using this criterion, we find for each row of the matrix A_{svd} the $N - M + \eta$ zero locations and consequently we can solve uniquely the system by selecting the $M - \eta$ nonzero elements for each row i.

Simulation results are provided and discussed in the following section.

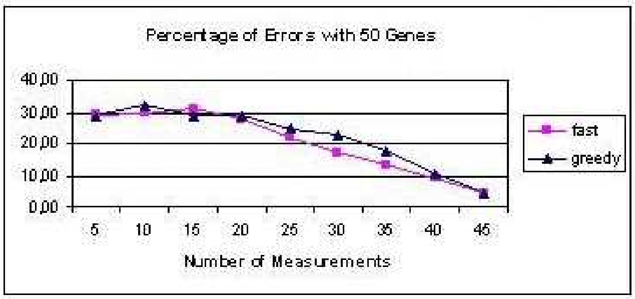


Figure 1: The percentage of errors estimated by the two algorithms vs. the number of available measurements (case of 50 genes). Data are generated with the linear model.

5 Results on artificial data

In this section, we will present the simulation results and the performances of the proposed reconstruction algorithm. First of all, we generate artificial data as described in the previous sections, then we estimate the matrix A, applying the reconstruction algorithms with $\eta = 1$. We measure the error by counting the percentage of sign discrepancies[6] $E = 100\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} e_{ij}}{N^2}$ where

$$e_{ij} = \begin{cases} 1 & \text{if sign}\,(a_{ij}) = \text{sign}\,(\bar{a}_{ij}) \\ 0 & \text{otherwise} \end{cases}$$

In Figure 1 we show the simulation results of a network composed by 50 genes. We have generated the data using the linear model. In the figure it is clear how both the algorithms we propose produce good results.

In Figure 2 we show the computing time[7] of the two algorithms. It is clear that the proposed *Fast* algorithm grows much less than *Greedy* search. If the number of the genes of the analyzed network increases (say 200, 500, 1000 or more genes) any exhaustive or greedy search, takes to much time to be computed. This is the strength of the algorithm we propose here.

[6] The information on the sign of the entries is the most relevant information from a biological point of view since reflect the nature of the gene-to-gene interaction.

[7] The algorithms described here have been fully implemented with Matlab. They ran on a Pentium IV with a clock speed of 1.8 GHz

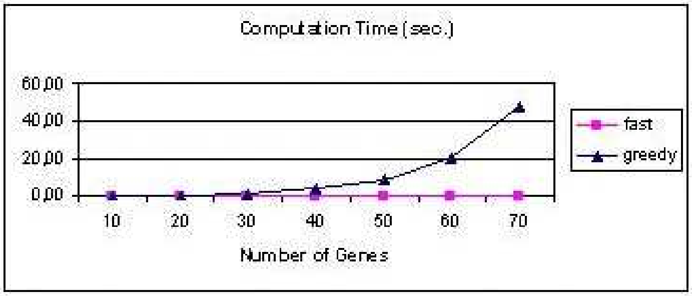


Figure 2: The computing time of the two algorithms vs. the number of genes in the network with $M = N/2$.

6 Results on real data

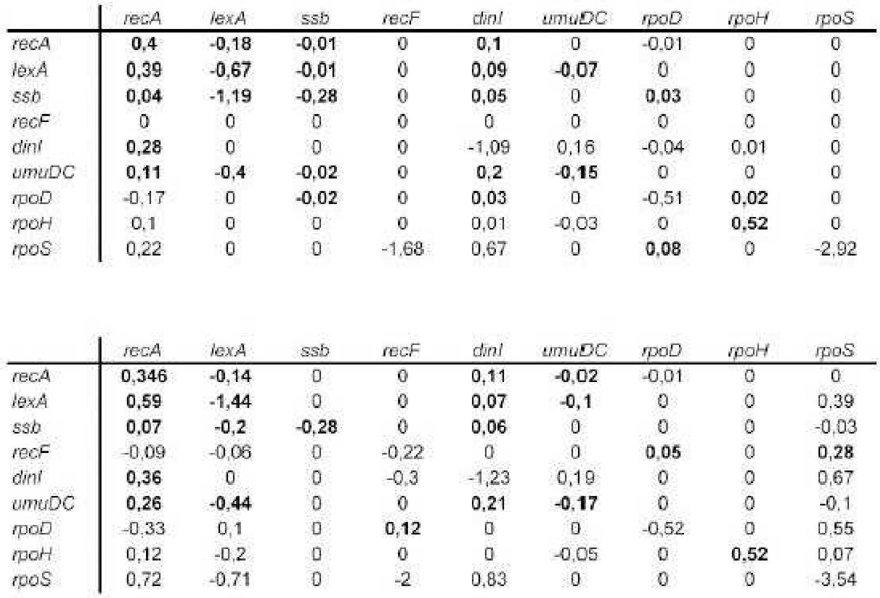
As an example of application of the algorithm proposed in this paper, we consider the data obtained in [2] for a nine-gene subnetwork of the SOS pathway in *Escherichia coli.* The collected data, *i.e.* the diagonal input matrix $B_{9\times 9}$ and the data matrix $Y_{9\times 9}$, is reported in the web supplement of the paper[8].

It is apparent from the the table in Figure 3, that the estimated matrix A performs quite well since the numbers of correct gene interaction signs (as reported by the biological literature on the SOS pathway in *E. coli*) is not very different from that of *Gardner et al.* It is very important to note that the approach used by *Gardner et al.* considers, for each row, all the possible positions of the 4 zeros in the dynamic matrix A, thus leading to a combinatorial approach since it evaluates every feasible solution. This is viable only for small subnetworks, while our method is suitable also for very large number of genes (*e.g.* the *Saccaromyces Cerevisiae* has about 6000 genes!) being polynomial in the number of genes.

7 Conclusions

In this paper we have addressed the so called ”reverse engineering” problem, that is the problem of reconstructing the gene-to-gene interactions from expression data (noisy samples of the state space trajectories) in the usual

[8] http://www.sciencemag.org/cgi/content/full/301/5629/102/DC1



	recA	*lexA*	*ssb*	*recF*	*dinI*	*umuDC*	*rpoD*	*rpoH*	*rpoS*
recA	**0,4**	**-0,18**	**-0,01**	0	**0,1**	0	-0,01	0	0
lexA	**0,39**	**-0,67**	**-0,01**	0	**0,09**	**-0,07**	0	0	0
ssb	**0,04**	**-1,19**	**-0,28**	0	**0,05**	0	**0,03**	0	0
recF	0	0	0	0	0	0	0	0	0
dinI	**0,28**	0	0	0	-1,09	0,16	-0,04	0,01	0
umuDC	**0,11**	**-0,4**	**-0,02**	0	**0,2**	**-0,15**	0	0	0
rpoD	-0,17	0	**-0,02**	0	**0,03**	0	-0,51	**0,02**	0
rpoH	0,1	0	0	0	0,01	-0,03	0	**0,52**	0
rpoS	0,22	0	0	-1,68	0,67	0	**0,08**	0	-2,92

	recA	*lexA*	*ssb*	*recF*	*dinI*	*umuDC*	*rpoD*	*rpoH*	*rpoS*
recA	**0,346**	**-0,14**	0	0	**0,11**	**-0,02**	-0,01	0	0
lexA	**0,59**	**-1,44**	0	0	**0,07**	**-0,1**	0	0	0,39
ssb	**0,07**	**-0,2**	**-0,28**	0	**0,06**	0	0	0	-0,03
recF	-0,09	-0,06	0	-0,22	0	0	**0,05**	0	**0,28**
dinI	**0,36**	0	0	-0,3	-1,23	0,19	0	0	0,67
umuDC	**0,26**	**-0,44**	0	0	**0,21**	**-0,17**	0	0	-0,1
rpoD	-0,33	0,1	0	**0,12**	0	0	-0,52	0	0,55
rpoH	0,12	-0,2	0	0	0	-0,05	0	**0,52**	0,07
rpoS	0,72	-0,71	0	-2	0,83	0	0	0	-3,54

Figure 3: The table (labelled with gene names) in the upper part of the picture is the identified matrix by Gardner et al., those in the lower part is obtained using our method. Numbers in bold indicate a correct sign. Note that a positive sign on the main diagonal denotes positive self-regulation apart from the self degradation rate, so that stability is guaranteed. The recF row must be disregarded since the experiments didn't yield reliable data.

case of "few" samples and "many" genes. The typical situation is $\sim$ 10 samples for gene networks with $\sim$ 5000 genes, as in [11].

We have proposed a *fast* algorithm for finding a "reasonable" solutions among the many feasible by exploiting the *a priori* biological information about sparseness of the gene networks. However, this hypothesis in too generic for selecting a "small" set of solutions, but it is of great help. The main problem is that we only know that there are "many" zeros, but no assumptions are made about the locations of such zeros. This makes even the problem of imposing sparseness a difficult problem since it has an intrinsic combinatorial complexity. To avoid this, in the literature many approaches have been developed mainly using a *greedy* search strategy, but still, the computation burden is very high. In this paper we have proposed and evaluated on artificial and real data, an algorithm with a very low computational complexity (linear in the number of genes) which can be used also for genome wide measurements. From this point of view, the algorithm presented seems very promising and future work will be devoted to explore its potential on large scale real data.

References

[1] M.K.S. Yeung, J. Tegner, and J.J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences*, 99:6163–6168, 2002.

[2] T.S. Gardner, D. di Bernardo, D. Lorentz, and J.J. Collins. Reverse engineering gene networks and identifying compound mode of action via expression profiling. *Science*, 301:102–105, 2003.

[3] T.Chen, H.L. He, and G.M. Church. Modeling gene expression with differential equations. *Proceedings of the Pacific Symposium on Biocomputing*, 4:29–40, 1999.

[4] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Linear modeling of mrna levels during cns development and injury. *Proceedings of the Pacific Symposium on Biocomputing*, 4:41–52, 1999.

[5] N.S. Holter, A. Maritan, M. Cieplak, N.V. Fedoroff, and J.R. Banavar. Dynamic modeling of gene expression data. *Proceedings of the National Academy of Sciences*, 98:1693–1698, 2001.

[6] E.P. van Someren, L.F.A.Wessels, and M.J.T. Reinders. Linear modeling of genetic networks from experimental data. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, pages 355–366, 2000.

[7] D.C. Weaver, C.T. Workman, and G.D. Stormo. Modeling regulatory networks with weight matrices. *Proceedings of the Pacific Symposium on Biocomputing*, 4:112–123, 1999.

[8] M.J.L. De Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano. Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. *Proceedings of the Pacific Symposium on Biocomputing*, 8:17–28, 2003.

[9] H. De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9:67–103, 2002.

[10] H. Jeong, B. Tomber, R. Albert, Z.N. Oltvai, and A.L. Barabasi. The large scale organization of metabolic networks. *Nature*, 407:651–654, 2000.

[11] A.P. Gasch, P.T. Spellman, C.M. Kao, O. Carmel-Harel, M.B. Eisen, G. Storz, D. Botstein, and P.O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–4257, 2000.